

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of:  
Steven T. Shaughnessy

Serial No.: 10/710,356

Filed: July 2, 2004

For: Database System Providing High  
Performance Database Versioning

Examiner: Sanders, Aaron J

Art Unit: 2169

APPEAL BRIEF

Mail Stop Appeal  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

**BRIEF ON BEHALF OF STEVEN T. SHAUGHNESSY**

This is an appeal from the Final Rejection mailed February 8, 2007, in which currently-pending claims 1-3, 5-8, 10-19, 21-24, and 26-30 stand finally rejected. Appellant filed a Notice of Appeal on June 11, 2007 (as indicated by return of a confirmation postcard marked "OIPE JUN 11 2007"). This brief is submitted electronically in support of Appellant's appeal.

## TABLE OF CONTENTS

1. REAL PARTY IN INTEREST .....	3
2. RELATED APPEALS AND INTERFERENCES .....	3
3. STATUS OF CLAIMS .....	3
4. STATUS OF AMENDMENTS .....	3
5. SUMMARY OF CLAIMED SUBJECT MATTER .....	4
6. GROUNDS OF REJECTION TO BE REVIEWED .....	9
7. ARGUMENT .....	10
A. First Ground: Claims 12 and 28 objected to as lacking antecedent basis .....	10
B. Second Ground: Claims 1, 12, 17 and 28 rejected under 35 U.S.C. 112, first paragraph .....	11
C. Third Ground: Claims 6, 8, 22 and 24 rejected under 35 U.S.C. 112, second paragraph .....	14
D. Fourth Ground: Claims 1-3, 5-8, 10-19, 21-24 and 26-30 rejected under 35 U.S.C. 101 .....	16
E. Fifth Ground: Claims 1-3, 5-8, 10, 12, 13, 15-19, 21-24, 26, 28 and 29 rejected under 35 U.S.C. 102(e) .....	20
F. Sixth Ground: Claims 14 and 30 rejected under 35 U.S.C. 103(a) .....	29
G. Seventh Ground: Claims 11 and 27 rejected under 35 U.S.C. 103(a) .....	32
8. CONCLUSION .....	33
9. CLAIMS APPENDIX .....	34
10. EVIDENCE APPENDIX .....	39
11. RELATED PROCEEDINGS APPENDIX .....	40

## **1. REAL PARTY IN INTEREST**

The real party in interest is assignee Borland Software Corporation, a Delaware corporation, located and doing business at 100 Enterprise Way, Scotts Valley, CA.

## **2. RELATED APPEALS AND INTERFERENCES**

There are no appeals or interferences known to Appellant, the Appellant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

## **3. STATUS OF CLAIMS**

Claims 1-3, 5-8, 10-19, 21-24, and 26-30 are pending in the subject application and are the subject of this appeal. Claims 4, 9, 20 and 25 were canceled. (No claims are allowed, confirmed, or withdrawn.) An appendix setting forth the claims involved in the appeal is included as the last section of this brief.

## **4. STATUS OF AMENDMENTS**

Two Amendments have been filed in this case. Appellant mailed an Amendment on January 3, 2007, in response to a non-final Office Action dated October 3, 2006. In the Amendment, the pending claims were amended in a manner which Appellant believes clearly distinguished the claimed invention over the art of record, for overcoming the art rejections. In response to the Examiner's Second Office Action dated February 8, 2007 (hereinafter "Second Office Action") finally rejecting Appellant's claims, Appellant filed an Amendment After Final dated May 8, 2007 which requested the entry of amendments to Appellant's specification and claims to address non-prior art objections and rejections made by the Examiner in the Second Office Action. Appellant's Amendment After Final also requested reconsideration of the prior art rejection of Appellant's claims. In an Examiner's Advisory Action mailed May 15, 2007 the Examiner refused to enter Appellant's Amendment After Final on the basis that the amended claims raised new issues which would require further search and consideration. In response to the

Examiner's Advisory Action, Appellant filed a Notice of Appeal. After filing the Notice of Appeal, Appellant and the Examiner conducted a telephone interview in which Appellant requested the Examiner to enter the amendments to the specification and claims made by Appellant in the Amendment After Final dated May 8, 2007 to address non-prior art rejections and narrow the issues for appeal. In response, the Examiner agreed to enter Appellant's amendments to the specification (only) in an Examiner action mailed July 19, 2007. However, the Examiner refused to enter the amendments to the claims requested by Appellant in the Amendment After Final dated May 8, 2007. Accordingly, no amendments to the claims have been entered in this case after the date of the Final Rejection.

## 5. SUMMARY OF CLAIMED SUBJECT MATTER

As to Appellant's **First Ground** for appeal, Appellant asserts that the Examiner's objection to claims 12 and 28 is improper, where the claimed invention is set forth in the embodiment in **independent claim 1**: In a database system employing a transaction log (Appellant's specification, paragraph [0016], paragraph [0047] (transaction log file(s)), paragraphs [0052]-[0053] (database system, log manager), paragraph [0060] (logging of transactions); Fig. 3 at 340 (database system) and at 350 (log manager); an improved method for restoring databases to a consistent version (Appellant's specification, paragraph [0016], paragraph [0047], paragraph [0071] (transactionally consistent view), paragraphs [0078]-[0088]; Fig. 4A at 401, 402, 403, 404, 405, Fig. 4B at 411, 412, 413; see also, paragraphs [0100]-[0103]); providing a shared cache storing database blocks for use by multiple databases (Appellant's specification, paragraph [0016], paragraph [0046], paragraph [0061], paragraphs [0071]-[0072]; Fig. 3 at 340 and at 360 (cache manager)); for a read-only transaction of a given database, creating a cache view of a given database using the given database's transaction log (Appellant's specification, paragraph [0016], paragraph [0047] (creating view in cache using transaction log), paragraphs [0060]-[0061], paragraphs [0078]-[0088]; Fig. 4A at 401, 402, 403, 404, 405; Fig. 4B at 411, 412, 413); said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time (Appellant's

specification, paragraph [0016], paragraph [0047] (using cache to create view of database at a particular point in time), paragraph [0071], paragraphs [0078]-[0082]; Fig. 4A at 402, 403, 404, 405; Fig. 4B at 411, 412, 413); creating a shadow cache for storing any database blocks that overflow said cache view during use of the cache view by the read-only transaction; (Appellant's specification, paragraph [0016], paragraph [0048] (if cache overflowing, shadow cache employed to hold blocks from old version that have been created), paragraphs [0049]-[0051], paragraph [0076], paragraphs [0078]-[0079], paragraph [0083], paragraph [0091]; Fig. 4A at 401); in conjunction with the cache view and the shadow cache, preserving a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit (Appellant's specification, paragraph [0016], paragraph [0051] (shadow cache used to preserve logical undo), paragraph [0076], paragraph [0083], Fig. 4A at 401, 404, 405; Fig. 4B at 411, 412, 413); and performing the logical undo operation in order to reconstruct a transactionally consistent prior version of the given database upon starting the read-only transaction (Appellant's specification, paragraph [0016], paragraph [0047], paragraph [0051] paragraph [0060], paragraph [0069], paragraph [0076], paragraph [0082], paragraph [0090] (once logical undo completed, cache view in transactionally consistent state); Fig. 4A at 405); thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses (Appellant's specification, paragraph [0047], paragraph [0071], paragraphs [0074]-[0076], paragraph [0083], paragraph [0090], Fig. 4A at 402, 403, 404, 405; Fig. 4B at 411, 412, 413). For Appellant's argument under the **First Ground** for appeal, Appellant additionally argues based on **dependent claim 12** which includes the limitation: reusing the cache view created for the read-only transaction for other read-only transactions, which start within a specified period of time following the start of the read-only transaction (see, e.g., Appellant's specification at paragraph [0072] providing that 5 different transactions can share the cache view).

Appellant further asserts that the Examiner's objection to claims 12 and 28 is improper, where the claimed invention comprises the embodiment set forth in **independent claim 17**: A database system capable of restoring databases to a consistent version (Appellant's specification, paragraph [0017], paragraph [0047], paragraphs

[0052]-[0053], paragraph [0060]; paragraph [0071], paragraphs [0078]-[0088]; Fig. 3 at 340; Fig. 4A at 401, 402, 403, 404, 405, Fig. 4B at 411, 412, 413); a log manager module which manages a transaction log of the database system (Appellant's specification, paragraph [0017], paragraph [0053], paragraph [0060](log manager module maintains and manages one or more logs); see particularly the log manager module at 350 at Fig. 3; a cache manager module (Appellant's specification, paragraph [0053], paragraph [0061] (cache manager manages cache); see particularly the cache manger module at 360 at Fig. 3); for managing a shared cache that stores database blocks for use by multiple databases (Appellant's specification, paragraphs [0016]-[0017], paragraph [0046], paragraph [0105]; see also Fig. 3 at 340, 360) and creating a cache view of a given database created using the transaction log of the given database (Appellant's specification, paragraph [0017], paragraph [0047] (creating view in cache using transaction log), paragraphs [0071]-[0072], paragraphs [0078]-[0082]; Fig. 4A at 401, 402, 403, 404, 405; Fig. 4B at 411, 412, 413); said cache view being created in response to a read-only transaction of the given database (Appellant's specification, paragraph [0017], paragraph [0047], paragraphs [0071]-[0072], paragraph [0078]; Fig. 4A at 402, 403), said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time (Appellant's specification, paragraph [0017], paragraph [0047] (using cache to create view of database at a particular point in time), paragraph [0071], paragraphs [0078]-[0082]; Fig. 4A at 402, 403, 404, 405; Fig. 4B at 411, 412, 413); wherein the cache manager utilizes a shadow cache for storing any database blocks that overflow said cache view during use of the cache view by the read-only transaction (Appellant's specification, paragraph [0017], paragraph [0048] (if cache overflowing, shadow cache employed to hold blocks from old version that have been created), paragraph [0061], paragraphs [0071]-[0072], paragraph [0083], paragraph [0091], paragraph [0105]; see also Fig. 3 at 340, 360); and a transaction manager module for performing read-only transactions of the database system (Appellant's specification, paragraph [0053], paragraphs [0059]-[0060]; see particularly, transaction manager module at 355 at Fig. 3); and which performs a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit (Appellant's specification,

paragraph [0017], paragraph [0051] paragraph [0060], paragraph [0076], paragraphs [0082]-[0083]; Fig. 4A at 405); in order to reconstruct a transactionally consistent prior version of the given database upon starting the read-only transaction thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses (Appellant's specification, paragraph [0017], paragraph [0071], paragraph [0074], paragraph [0090] (once logical undo completed, cache view in transactionally consistent state); Fig. 4A at 403, 404, 405; Fig. 4B at 411, 412, 413). For Appellant's argument under the **First Ground** for appeal, Appellant additionally argues based on **dependent claim 28** which includes the limitation: wherein said cache manager reuses the cache view created for the read-only transaction for other read-only transactions which start within a specified period of time following the start of the read-only transaction (see, e.g., Appellant's specification at paragraph [0072] providing that 5 different transactions can share the cache view).

As to Appellant's **Second Ground** for appeal, Appellant asserts that the Examiner's rejection of claims 1, 12, 17 and 28 under **35 U.S.C. 112, first paragraph**, as failing to comply with the enablement requirement is improper, where the claimed invention is set forth in the embodiment in **base independent claims 1 and 17** (the mapping of which is shown above under Appellant's First Ground for appeal, and which hereby is incorporated by reference). For Appellant's argument under the **Second Ground** for appeal, Appellant additionally argues based on **dependent claims 12 and 28** (the mapping of which is also shown above under Appellant's First Ground for appeal, and which hereby is incorporated by reference).

As to Appellant's **Third Ground** for appeal, Appellant asserts that the Examiner's rejection of claims 6, 8, 22, and 24 under **35 U.S.C. 112, second paragraph**, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Appellant regards as the invention, where the claimed invention is set forth in the embodiment in **base independent claims 1 and 17** (the mapping of which is shown above under Appellant's First Ground for appeal, and which hereby is incorporated by reference). For Appellant's argument under the **Third Ground** for appeal, Appellant additionally argues based on **dependent claims 6 and 22** which include claim limitations of an allocation bitmap for indicating database blocks in use in the shadow cache

(Appellant's specification, paragraph [0079], paragraph [0097]). For Appellant's argument under the **Third Ground** for appeal, Appellant additionally argues based on **dependent claims 8 and 24** which include claim limitations of a shadow cache comprising a temporary database table including a first column for maintaining a block number of a cache view block having undo/redo records applied to it and a second column for maintaining a block number in a temporary database allocated to save off a modified block from the cache views (Appellant's specification, paragraphs [0048]-[0049] and particularly, in paragraph [0079]).

As to Appellant's **Fourth Ground** of appeal as to whether claims 1-3, 5-8, 10-19, 21-24 and 26-30 are unpatentable under 35 USC Section 101 as directed to non-statutory subject matter, where the claimed invention is set forth in the embodiment in **base independent claims 1 and 17** (the mapping of which is shown above under Appellant's First Ground for appeal, and which hereby is incorporated by reference).

As to Appellant's **Fifth Ground** of appeal, Appellant asserts that the art rejection under **Section 102(e)** relying on U.S. Patent No. 5,715,447 issued to Hayashi et al. ("Hayashi") fails to teach or suggest all of the claim limitations of Appellant's rejected claims 1-3, 5-8, 10, 12, 13, 15-19, 21-24, 26, 28 and 29, where the claimed invention is set forth in the embodiment in **base independent claims 1 and 17** (the mapping of which is shown above under Appellant's First Ground for appeal, and which hereby is incorporated by reference).

As to Appellant's **Sixth Ground** for appeal, Appellant asserts that the art rejection under **Section 103(a)** relying on the combination of Hayashi (above) and U.S. Patent No. 5,701,480 issued to Raz ("Raz") fails to teach or suggest all of the claim limitations of Appellant's rejected claims 14 and 30, where the claimed invention is set forth in the embodiment in **base independent claims 1 and 17** (the mapping of which is shown above under Appellant's First Ground for appeal, and which hereby is incorporated by reference). For Appellant's argument under the **Sixth Ground** for appeal, Appellant additionally argues based on **dependent claims 14 and 30** which includes the limitation: if the read-only transaction must be undone, using the back link log records to skip portions of the transaction log that are irrelevant for undoing the read-only transaction, wherein the back link log records are only generated in the transaction log when there are



active read only transactions (Appellant's specification, paragraph [0093]).

As to Appellant's **Seventh Ground** for appeal, Appellant asserts that the art rejection under **Section 103(a)** relying on the combination of Hayashi and *The Authoritative Dictionary of IEEE Standard Terms, 7th Edition*, IEEE Press, 2000 ("IEEE") fails to teach or suggest all of the claim limitations of Appellant's rejected claims 11 and 27, where the claimed invention is set forth in the embodiment in **base independent claims 1 and 17** (the mapping of which is shown above under Appellant's First Ground for appeal, and which is hereby incorporated by reference).

## **6. GROUNDS OF REJECTION TO BE REVIEWED**

The grounds for appeal are:

(1st) Whether claims 12 and 28 are properly objected to as lacking antecedent basis in the specification.

(2nd) Whether claims 1, 12, 17 and 28 are unpatentable under 35 U.S.C. Section 112, first paragraph as failing to comply with the enablement requirement.

(3rd) Whether claims 6, 8, 22, and 24 are unpatentable under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Appellant regards as the invention.

(4th) Whether claims 1-3, 5-8, 10-19, 21-24 and 26-30 are unpatentable under 35 USC Section 101 as directed to non-statutory subject matter.

(5th) Whether claims 1-3, 5-8, 10, 12, 13, 15-19, 21-24, 26, 28 and 29 are unpatentable under 35 U.S.C. 102(c) as being anticipated by U.S. Patent No. 5,715,447 issued to Hayashi et al. (hereinafter "Hayashi");

(6th) Whether claims 14 and 30 are unpatentable under 35 U.S.C. 103(a) as being obvious over Hayashi in view of U.S. Patent No. 5,701,480 issued to Raz (hereinafter "Raz"); and

(7th) Whether claims 11 and 27 are unpatentable under 35 U.S.C. 103(a) as being obvious over Hayashi in view of *The Authoritative Dictionary of IEEE Standard Terms, 7th Edition*, IEEE Press, 2000 (hereinafter "IEEE");

## 7. ARGUMENT

### A. First Ground: Claims 12 and 28 objected to as lacking antecedent basis

Claims 12 and 28 are objected to by the Examiner on the basis that the limitation "reusing the cache view" lacks antecedent basis in the disclosure (Second Office Action, page 3). Appellant respectfully disagrees with the Examiner as Appellant's specification specifically describes a read-only transaction delay setting which allows the same cache view to be used (reused) by a plurality of read-only transactions which start within a user-configurable time interval from the start of a given read-only transaction for which the cache view was created. This is included, for example, in paragraph [0072] of Appellant's specification which provides as follows:

Read-only transactions have a property called the read-only transaction delay setting. This setting enables one to specify, for example, a 30 second delay and if five read-only transactions are all started within the same period (e.g., the same 30 second period) they can share the same cache view. As a result, five different read-only cache views are not required for the five transactions. This setting can be user configured. For instance, if a user does not care if a transaction is out of date by ten minutes then it can be set for ten minutes. In this case, every transaction that starts within that same ten-minute bracket will share the same cache view.

(Appellant's specification, paragraph [0072], emphasis added)

Appellant respectfully believes that the above describes that the cache view created for use by one read-only transaction may be reused by other read-only transactions. Appellant also notes that Appellant requested entry of an amendment to Claims 12 and 28 in Appellant's Amendment after Final to replace the word "reusing" in these claims with the word "sharing", so that the claim limitation would state "sharing the cache view" which more closely matches the wording used in Appellant's specification. However, the Examiner refused to enter Appellant's amendment to these claims in the Amendment after Final. For the reasons set forth above, it is respectfully submitted that the Examiner's objection to claims 12 and 28 as lacking antecedent basis in the Appellant's specification is improper.

**B. Second Ground: Claims 1, 12, 17 and 28 rejected under 35 U.S.C. 112, first paragraph**

**1. General**

Claims 1, 12, 17, and 28 stand rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The first paragraph of 35 U.S.C. Section 112 provides as follows:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

As noted in *In re Wertheim*, 541 F.2d 257, 263, 191 USPQ 90, 97 (CCPA 1976), there is a strong presumption that an adequate written description of the claimed invention is present when the application is filed. As will be shown below, Appellant's claims 1, 12, 17, and 28 comply with the written description requirement of the first paragraph of 35 U.S.C. Section 112 as they are expressly supported in Appellant's specification.

**2. Claims 1 and 17**

Claims 1 and 17 stand rejected under 35 U.S.C. Section 112, first paragraph on the basis that the recited limitation of a "cache view" is not enabled. The following is the Examiner's rationale for rejection of Appellant's claims 1 and 17 under 35 U.S.C. Section 112, first paragraph:

As per claims 1 and 17, the recited "cache view" is not enabled. In the instant claims, a "cache view" is defined as "comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time". This appears to be a "snapshot" of the database and is not consistent with the definition of a cache. Applicant further defines a "cache view" in paragraph [0047] of the instant specification by saying that it is a process in which the transaction log is combined with the blocks of data in the cache to create a version of the database at an earlier state. This appears to be a rollback routine. Applicant discusses "cache views" as if they are objects, but has defined them as if they are processes. It is not clear how an object can also be a process.

(Second Office Action, page 3)

Appellant respectfully disagrees with the Examiner that the "cache view" limitation of Appellant's claims 1 and 17 is not enabled. Appellant's claims provide that a "cache view" is created as part of the methodology of Appellant's claimed invention (see e.g., Appellant's claim 1). The "cache view" which is created is a view of the database a particular point in time which is created by logically undoing transactions (e.g., active transactions that have not committed at the time the read-only transaction starts) so as to create a version of the database that is transactionally consistent as of a particular point in time (Appellant's specification, paragraphs [0070]-[0071], paragraph [0076]). This view that is created is maintained in cache (i.e., the database blocks or pages are maintained in system memory) for use in performing a read-only transaction(s) (see, e.g., paragraph [0072] of Appellant's specification set forth above which describes creating a cache view which is shared by all transactions starting within a 30 second interval). The cache view is created by logically undoing active transactions at the start of a read-only transaction as hereinafter discussed in more detail.

By creating the cache view for a read-only transaction, Appellant's invention avoids the need for a transactional clean point (Appellant's specification, paragraph [0076]). However, as performing the logical undo to create this view requires the modification of some database blocks, the system needs to hold these modified blocks in cache until the read-only transaction commits (Appellant's specification, paragraph [0076]). Thus, Appellant refers to this view of the database which is maintained in cache for use by read-only transactions in numerous paragraphs of Appellant's specification as either a "cache view" or a "read-only cache view" (see e.g., paragraphs [0016], [0017], [0022], [0047], [0050], [0071], as well as others). For example, paragraph [0047] of Appellant's specification provides as follows:

With read-only transactions the methodology of the present invention provides for introducing another cache view. This new cache view is for the same database as the write cache view but it is a separate view of the database. In general terms, it is a view of a version of a database at a particular point in time. The approach of the present invention involves using the cache in order to create a view of the database at a particular point in time. Since details regarding all changes to the database are recorded in the transaction log file(s), the transactional log file(s) may be used to reconstruct a view of the database at this particular point in time. The view is constructed by applying log records to this view in the cache, so as to

create a version of the database at a particular point in time.

(Appellant's specification, paragraph [0047], emphasis added)

In addition, Appellant's claims themselves include language describing the meaning of the term "cache view". For instance, Appellant's claim 1 includes the following limitations:

for a read-only transaction of a given database, creating a cache view of a given database using the given database's transaction log, said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time;

(Appellant's claim 1, emphasis added)

As shown above, Appellant's claims include a clear description of the meaning of the term "cache view" which is consistent in all respects with the description of this term set forth in Appellant's specification. Moreover, it is well settled that patent law allows an inventor to be his own lexicographer (see e.g., *In re Paulsen*, 30 F.3d 1475, 1480, 31 USPQ2d 1671, 1674 (Fed. Cir. 1994)). Although Appellant believes that the term "cache view" is appropriate given that Appellant is referencing a "view" of the database which is maintained in cache, to the extent that this might conceivably depart from the normal meaning of these terms, Appellant has made it clear in both the Appellant's specification and claims the meaning the Appellant attributes to the term "cache view". Thus, Appellant respectfully believes that these claims are supported both in Appellant's specification and in Appellant's claims themselves, making the meaning of the term clear and understandable to those skilled in the art.

### 3. Claims 12 and 28

Claims 12 and 28 also stand rejected under 35 USC 112, first paragraph as failing to comply with the enablement requirement. As to these claims, the Examiner states that there does not appear to be any indication in Appellant's specification of how the "cache view" is "reused". As described above in the discussion of Appellant's First Ground of appeal (which discussed the Examiner's objection to these same claims as lacking antecedent basis), paragraph [0072] of Appellant's specification includes specific description of a read-only transaction delay setting which enables one to specify, for

example, that five read-only transactions started within a 30 second period can reuse (or share) the same cache view. This delay setting features avoids the need to create and maintain in cache a separate view for each read-only transaction, which would use additional resources and may potentially cause cache overflow. Instead, the cache view created for a given read-only transaction is made available for use (i.e., reuse) by other read-only transactions starting within a user-configurable time period of the given read-only transaction. For the reasons discussed above in the First Ground of appeal (which are incorporated herein by reference), Appellant respectfully believes that the claim limitations of claims 12 and 28 are specifically described in the Appellant's specification.

#### 4. Conclusion

For the reasons set forth above, it is respectfully submitted that the Examiner's rejection of claims 1, 12, 17 and 28 under 35 USC 112, first paragraph as failing to comply with the enablement requirement should be not be sustained.

### **C. Third Ground: Claims 6, 8, 22 and 24 rejected under 35 U.S.C. 112, second paragraph**

#### 1. General

Claims 6, 8, 22, and 24 stand rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Appellant regards as the invention. As will be shown below, Appellant's claims 6, 8, 22 and 24 comply with the requirements of the second paragraph of 35 U.S.C. Section 112 as they point out and distinctly claim the subject matter of Appellant's invention.

#### 2. Claims 6 and 22

The Examiner rejected claims 6 and 22 on the basis that these claims disclose "an allocation bitmap"; however, claim 6 is a method claim and therefore cannot "provide" a device (Second Office Action, page 4). Appellant's claim 1 includes claim limitations of a shadow cache which is used as a backing store for storing any database blocks that overflow the cache during use of the cache view by a read-only transaction. Appellant's claim 6 adds additional limitations of an allocation bitmap which indicates database blocks which are in use in the shadow cache. Appellant's claim 6 provides as follows:

The method of claim 1, further comprising:

providing an allocation bitmap for indicating database blocks in use in the shadow cache.

(Appellant's claim 6)

Appellant believes the meaning of the above claim 6 is clear as it provides that the database blocks used in the shadow cache are tracked using an allocation bitmap. The Examiner's stated concern is with the use of the word "providing" in claim 6 which is a method claim. Although Appellant believes the meaning of the claim to be clear, in Appellant's Amendment after Final, Appellant requested the Examiner to enter an amendment to replace the word "providing" with the word "using" so as to address the Examiner's concern. However, the Examiner refused to enter this amendment.

The Examiner has similarly rejected claim 22 which includes claim limitations of a cache manager maintains an allocation bitmap indicating database blocks in use in the shadow cache. As to claim 22, Appellant respectfully believes the Examiner's rationale for rejection is not applicable as claim 22 is not a method claim and it does not use the word "providing". Thus, as no other rationale for rejection of claim 22 is provided, Appellant respectfully believes that the rejection of claim 22 should not be sustained.

### 3. Claims 8 and 24

The Examiner also rejected claims 8 and 24 under 35 USC 112, second paragraph on the basis that meaning of the phrase "to save off" is not apparent in the context of the claim limitations of Appellant's claims 8 and 24 (Second Office Action, page 4). Appellant's claim 8, for example, includes the following claim limitations:

The method of claim 1, wherein the shadow cache comprises a temporary database table including a first column for maintaining a block number of a cache view block having undo/redo records applied to it and a second column for maintaining a block number in a temporary database allocated to save off a modified block from the cache view.

(Appellant's claim 8)

Recall that the shadow cache is employed in Appellant's invention as a backing store to hold database blocks of the cache view created for a read-only transaction that overflow the cache (see e.g., Appellant's claim 1 and prior discussion above). As the

cache view is created by logically undoing active transactions, modifications to database blocks may be made in creating the cache view. These modified blocks need to be retained for use by the read-only transaction until the read-only transaction commits (Appellant's specification, paragraph [0076]). As these modifications cannot be written back to the database (i.e., as they may corrupt the database given that they relate to a different version of the database) the shadow cache is used, if needed, for temporarily saving these modified blocks during the duration of the read-only transaction (Appellant's specification, paragraph [0048]).

As illustrated above, Appellant's claim 8 simply adds claim limitations that the shadow cache includes a temporary database table having two columns for mapping blocks from the cache view to the backing store in the temporary database (Appellant's specification, paragraph [0048]). (Appellant's claim 24 includes similar limitations.) The first column maintains a block number of a cache view block which had undo or redo records applied to it and the second indicates the block number in the temporary database where the modified block is saved off (i.e., saved) (Appellant's specification, paragraph [0048]). Appellant respectfully believes that the meaning is clear in that the claim limitations simply reference a block number of the temporary database in which a particular block is saved. In addition, Appellant notes that the limitation of "a block number in a temporary database allocated to save off a modified block" is specifically included at paragraph [0048] of Appellant's specification.

#### 4. Conclusion

For the reasons set forth above, it is respectfully submitted that the Examiner's rejection of claims 6, 8, 22, and 24 under 35 USC 112, second paragraph as indefinite should be not be sustained.

### **D. Fourth Ground: Claims 1-3, 5-8, 10-19, 21-24 and 26-30 rejected under 35 U.S.C. 101**

#### 1. General

Claims 1-3, 5-8, 10-19, 21-24, and 26-30 stand rejected under 35 USC Section 101 as directed to non-statutory subject matter. The Examiner states that Appellant's claims are directed to a method and system for restoring databases to a consistent version. which lacks a practical application of a judicial exception (law of nature, abstract idea,



naturally occurring article/phenomena) since it fails to produce a tangible result.

Article 1, Section VIII of the U.S. Constitution authorizes patent protection for inventions that promote the progress of the useful arts. Congress acted upon this authorization in 35 USC Section 101 by mandating patent protection for inventions without regard to the form or physical embodiment of the invention as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

(35 USC Section 101)

Congress intended that "anything under the sun that is made by man" should be patentable, with the exception of "laws of nature, physical phenomena and abstract ideas." *Diamond v. Chakrabarty*, 447 U.S. 303, 309 (1980); *Diamond v. Diehr*, 450 U.S. 175, 182 (1981). As noted in *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992), the Examiner bears the initial burden of presenting a *prima facie* case of unpatentability under Section 101. As will be shown below, the Examiner has failed to meet this burden as Appellant's claimed invention is directed to statutory subject matter and provides useful, concrete and tangible results.

## 2. Claims 1-3, 5-8, 10-16

As to claims 1-3, 5-8, and 10-16, the Examiner rejects these claims under 35 USC Section 101 on the basis that the claimed subject matter does not produce a tangible result as follows:

...the claimed subject matter fails to produce a result that is limited to having real world value rather than a result that may be interpreted to be abstract in nature as, for example, a thought, a computation, or manipulation of data. More specifically, the claimed subject matter provides for "returning a result comprising a transactionally consistent version of the given database supporting read-only uses". This produced result remains in the abstract and, thus, fails to achieve the required status of having real world value.

(Second Office Action, page 5)

At the outset, Appellant's claims clearly fall within one or more of the statutory categories set forth in Section 101 as they comprise a "new and useful process, machine,

manufacture or composition of matter" (or a new and useful improvement thereof). Thus, as Appellant respectfully believes that the claimed invention falls within the statutory categories of invention set forth in Section 101, they are patentable unless they comprise "laws of nature, physical phenomena and abstract ideas." In addition, while abstract ideas, natural phenomena, and laws of nature are not eligible for patenting, methods and products employing abstract ideas, natural phenomena, and laws of nature to perform a real-world function may be patentable.

As described above in the Second Ground for appeal (which is hereby incorporated by reference), Appellant's claimed invention provides for creating a view of a database that is transactionally consistent as of a particular point in time (Appellant's specification, paragraphs [0070]-[0071], paragraph [0076]). This view that is created is maintained in cache (i.e., in system memory) for use in performing read-only transactions (see, e.g., paragraph [0072] of Appellant's specification). These features of creating a view of the database at a particular point in time for performing read-only transactions are recited as limitations of Appellant's claims. For example, Appellant's claim 1, including the following claim limitations:

for a read-only transaction of a given database, creating a cache view of a given database using the given database's transaction log, said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time; ...  
performing the logical undo operation in order to reconstruct a transactionally consistent prior version of the given database upon starting the read-only transaction, thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses.

(Appellant's claim 1)

By creating and maintaining in memory a view of the database as of a given point in time, Appellant's invention enables read-only transactions to be performed in a manner that does not block other transactions (e.g., write transactions) being performed against the database. Appellant's invention also avoids the need for a transactional clean point (Appellant's specification, paragraph [0076]), thereby improving performance of the database system. Thus, Appellant respectfully believes that Appellant's claimed invention produces useful, concrete and tangible results.

### 3. Claims 15 and 16

The Examiner additionally rejects claim 15 stating which includes claim limitations of a "computer-readable medium having processor-executable instructions" on the basis that the instructions may not have been executed and rejects claim 16 on the basis that the "downloadable set of processor-executable instructions" may have not been downloaded and could be downloaded on a non-statutory carrier wave (Second Office Action, page 5). Appellant respectfully believes that claims 15 and 16, which are dependent upon claim 1 overcome the rejection under Section 101 for the reasons stated above with respect to claim 1.

### 4. Claims 17-19, 21-24, and 26-30

As per claims 17-19, 21-24, and 26-30, the Examiner has rejected these claims on the basis that Appellant's system does not require any hardware, therefore making it software *per se* and, as such, non-statutory (Second Office Action, page 5). Initially, even assuming that Appellant's claimed invention is implemented in software, Appellant does not agree that software falls outside of Section 101 as not comprising a new and useful process, machine, manufacture, or composition of matter under the standards described above.

In addition, Appellant respectfully believes that the Examiner has incorrectly construed Appellant's specification and claims as indicating that the elements of Appellant's invention are implemented solely in software. In fact, Appellant does not believe that Appellant's claims 17-19, 21-24 and 26-30 can reasonably be construed as comprising software *per se*. As described above (in the discussion of the rejection of claims 1-3, 5-8, 10-16 under Section 101), Appellant's claimed invention produces and maintains in memory (i.e., cache memory of a computer system) a view of the database which may be used for performing read only transactions. Appellant's claim 17 includes similar limitations. By definition, a "cache" is a type of computer memory that contains recently accessed data, designed to speed up subsequent access to the same data (Appellant's specification, paragraph [0061]). Thus, as maintaining database blocks in memory (e.g., in the cache and/or the shadow cache) requires hardware of some sort, Appellant's claimed invention clearly includes elements other than software. Similarly, Appellant's claim 17 provides the cache view is created and maintained in memory in

response to a read-only transaction and is utilized for read-only purposes (i.e., performing read-only database transactions).

Appellant's specification also expressly states that the elements of the invention may be implemented in hardware, software or firmware or combinations thereof. This is expressly stated, for example, in Appellant's specification as follows: "...the corresponding apparatus element may be configured in hardware, software, firmware or combinations thereof" (Appellant's specification, paragraph [0033], emphasis added).

Appellant's specification also describes in detail a computer hardware and software environment in which Appellant's invention may be implemented (Appellant's specification, paragraphs [0035]-[0045]). Moreover, Appellant states that the software (computer-executable instructions) direct operation of a device under processor control, such as a computer (Appellant's specification, paragraph [0063]). As Appellant's claim defines a useful machine or item of manufacture in terms of a hardware or hardware and software combination, Appellant respectfully believes that it comprises statutory subject matter.

#### 5. Conclusion

For the reasons set forth above, it is respectfully submitted that the Examiner's rejection of Claims 1-3, 5-8, 10-19, 21-24, and 26-30 under 35 USC Section 101 as directed to non-statutory subject matter should be not be sustained.

### **E. Fifth Ground: Claims 1-3, 5-8, 10, 12, 13, 15-19, 21-24, 26, 28 and 29 rejected under 35 U.S.C. 102(e)**

#### 1. General

Under Section 102, a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in the single prior art reference. (See, e.g., MPEP Section 2131.) As will be shown below, the Hayashi reference fails to teach each and every element set forth in independent claim 1 (and 45), as well as other claims, and therefore fails to establish anticipation of the claimed invention under Section 102.

#### 2. Claims 1-3, 5-8, 10, 12, 13, 15-19, 21-24, 26, 28 and 29

Claims 1-3, 5-8, 10, 12, 13, 15-19, 21-24, 26, 28 and 29 stand rejected under 35

U.S.C. 102(c) as being anticipated by U.S. Patent No. 5,715,447 issued to Hayashi et al. (hereinafter "Hayashi"). The following rejection of Appellant's claim 1 by the Examiner is representative of the Examiner's rejection of the Appellant's claims as being anticipated by Hayashi:

1. (Currently amended) In a database system employing a transaction log, an improved method for restoring databases to a consistent version, the method comprising:

providing a shared cache storing database blocks for use by multiple databases (See e.g. Fig. 4, "shared buffers 17");

for a read-only transaction of a given database, creating a cache view of a given database using the given database's transaction log, said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time (See e.g. Fig. 4, "log buffer 16" where, see col. 2, lines 20-27, "a log buffer for temporarily storing pre-update and post-update logs");

creating a shadow cache for storing any database blocks that overflow said cache view during use of the cache view by the read-only transaction (See e.g. Fig. 2 where, see col. 4, lines 8-18, "A buffer shared by the transactions is a bit map 30. The database 20 includes overflow pages 31. A database 20' is used to nonvolatilize the contents of the shared buffer. The bit map 30 controls overflow pages 31");

in conjunction with the cache view and the shadow cache, preserving a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit (See e.g. col. 5, lines 30-46, "When the contents of the updated page are completely written back to the allotted page on the disk, an original page is shifted to the allotted page in the table, and then a commitment is given to the transaction. A rollback is achieved by simply discarding the allotted page". It is noted that [0102] of Applicant's specification states, "For example, physical redo, physical undo, and logical undo are all concepts that exist in log-based transaction management systems" and are therefore admitted prior art); and

performing the logical undo operation in order to reconstruct a transactionally consistent prior version of the given database upon starting the read-only transaction, thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses (See e.g. col. 5, lines 30-46, "When the contents of the updated page are completely written back to the allotted page on the disk, an original page is shifted to the allotted page in the table, and then a commitment is given to the transaction. A rollback is achieved

by simply discarding the allotted page").

(Second Office Action, pages 6-7)

As noted above, a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in the single prior art reference. As will be shown below, Hayashi fails to teach each and every element set forth in independent claims 1 and 17 (as well as other claims) and therefore fails to establish anticipation of the claimed invention under Section 102.

The Examiner equates Hayashi's system and methodology for accelerating throughput, which is largely applicable for write transactions during online transaction processing (OLTP), such as an online reservation system where individual records are retrieved at random, to Appellant's invention which creates a read-only view in cache so as to keep read-only transactions from being blocked by exclusive transactional locks of write transactions (and also avoids write transactions being blocked by long duration read transactions). Thus, Appellant's claimed invention primarily relates to DSS (decision support system) uses, such as reporting, where records are sequentially retrieved in large blocks while Hayashi's is focused on OLTP transaction processing.

In addition to the different focus of Appellant's invention and Hayashi's solution, Appellant's claimed invention also includes specific elements that distinguish it from Hayashi's system. Hayashi's approach, essentially describes an optimization for write transactions, specifically "check pointing." Hayashi describes optimization techniques for reading and writing share blocks, especially the optimization of buffer locks used during nonvolatization. Hayashi describes the making of "copies" of buffers updated by write transactions during their process of nonvolatization. Appellant's approach, in contrast, provides a mechanism to reconstruct a transactionally consistent read-only view of the database by logically undoing all incomplete transactions at the point in time when the read-only version or view is to be constructed (Appellant's specification, paragraphs [0082]-[0083]). In a given database environment, one may have multiple database users with intermingled write operations occurring against the database, some of which have committed while others are still pending. Appellant's approach provides a mechanism to reconstruct a transactionally consistent read-only view of the database in cache which

may be used read-only transactions without blocking the write operations which may be concurrently occurring against the database. These and other differences between Appellant's invention and Hayashi's system become apparent when the elements of Appellant's claims are compared to the specific teachings of Hayashi cited by the Examiner.

Appellant's approach provides for creating a cache view of a given database using the database's transaction log. This is, for example, included as limitations of Appellant's claim 1 as follows:

for a read-only transaction of a given database, creating a cache view of a given database using the given database's transaction log, said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time;

(Appellant's claim1)

As described above in detail in Appellant's Second Ground for appeal (which is incorporated herein by reference), the cache view comprises particular database blocks (maintained in the shared cache and/or the shadow cache) that record a view of a particular version of the database at a given point in time. The Examiner references Hayashi's "log buffer 16" shown at Fig. 4 as well as col. 2, lines 20-27 of the Hayashi specification as providing the corresponding teachings. However, the referenced portion of Hayashi makes no mention of creating a version (or view) of a database at a given point in time using the transaction log. Instead, Hayashi merely describes a log buffer as follows:

A data processing system according to another aspect of the present invention includes, but is not limited to, a database management system for accessing and managing a database, at least one buffer shared by transactions, a log buffer for temporarily storing pre-update and post-update logs, a log file for storing the pre-update and post-update logs, and the database for storing data.

(Hayashi, col. 2, lines 20-27)

As illustrated above, Hayashi simply describes a log buffer for temporarily storing pre-update and post-update logs. This is not at all comparable to Appellant's invention which provides for using a database's transaction log to create a transactionally consistent

view of the database at a particular point in time which is maintained in memory for use by a read-only transaction.

Appellant's invention also includes a "shadow cache" which is used, if necessary, for maintaining in memory any database blocks of the cache view of the database which overflow the cache. As previously described, the shadow cache is implemented as a temporary database maintained in memory as a backing store in the event that the cache view (i.e., the transactionally consistent view of the database at a given point in time created for use by a read-only transaction) overflows the cache. The Examiner references Hayashi at col. 4, lines 8-18 for the corresponding teachings. However, Hayashi simply describes writing the contents of a shared buffer to a database so as to nonvolatilize the contents of the shared buffer.

Another important aspect of Appellant's invention is the use of "logical undos" for reconstructing a transactionally consistent (i.e., proper, as of a given point in time) version of the database. In order to understand this aspect better, it is helpful to review undo operations. Physical undo and redo operations can be achieved by simply replaying the database log back and forth. A logical undo operation, however, must happen at a higher level -- that is, at the basic top level that records are added or dropped. Physical level, on the other hand, is more concerned about moving bytes, for example in the same manner that edits would occur in a text editor. In other words, in a physical undo or redo operation, the system need only move bytes around, for example undoing a change by copying prior contents (i.e., restoring a record by copying a prior byte sequence back into the record). And certainly in situations where possible, physical undo and physical redo operations provide an easy (if not the easiest) approach for undoing or redoing changes. However, in a multi-user database environment, it is often not possible to simply use physical undo and redo.

Consider two database users: User A and User B. Although both users typically will not be modifying the same database record at the same time, they will often be modifying the same database table, say an ORDER ENTRY table. Now, consider a scenario where a read-only transaction has begun (e.g., for generating a report or performing a data mining operation) and User A's changes to the ORDER ENTRY table have committed, but they are unfortunately intermingled with User B's changes that have



yet to commit. Since the database log (i.e., data structure tracking the foregoing changes) has recorded physical changes to the database table in sequential order, one cannot simply undo B's changes without undoing A's changes. In other words, one cannot physically undo B's changes in a manner that will not affect A's changes.

In this scenario, the log also includes "logical" log records. These are high-level records that record logical (high-level) database operations, that is, ones that may be viewed as occurring at a top level, such as adding or dropping a database record. Note in particular that a high-level logical change to the table (e.g., adding a record) typically entails many physical changes besides the physical changes to the table itself (e.g., adding a byte sequence representing record data). For example, as a result of adding a record to a table any index on that table will likely need to be updated, thus requiring a physical change to affected indexes. Given the potential for a multitude of propagating changes or dependencies that may occur as a result of a single high-level change to a table, it is therefore helpful to represent these changes as high-level logical operations. In other words, a single high-level logical operation may be used to conveniently represent multiple lower-level physical operations.

The notion of a "logical undo" provides, in a similar manner, a higher-level representation of an undo (e.g., undo the insertion of a new record). This provides a higher-level representation than the multitude of physical undo operations that would need to be tracked to correctly represent the physical undoing of the operation against the entire database (i.e., physical undo of byte sequence change at the table, physical undo of the byte sequence change at the index, and so forth and so on).

Appellant's claims are not directed to just physical undo and redo operations, but instead are directed at a higher level: logical undo operations. As described above, in a given database environment, one may have multiple database users with intermingled write operations occurring against the database, some of which have committed while others are still pending. The notion of logical undo supports Appellant's approach of getting the database to a transactionally consistent prior state that is suitable for performing long-duration read-only transactions (e.g., DSS, reporting, data mining, or the like). Importantly, Appellant's logical undo approach provides a mechanism to separate the ones that have committed from those that have not.

In accordance with Appellant's invention, a read-only transactionally consistent version is reconstructed by **logically undoing all incomplete transactions** at that point in time (i.e., at the point in time when the read-only version or view is to be reconstructed). Here, transactions that were started but unfinished (at the time that the read-only transactionally consistent view is to be reconstructed) are logically undone. As a practical matter, all of these transactions include intermingled changes and therefore cannot simply be physically undone. If they could be physically undone, then the simplest approach would be to just proceed with physically undoing them. As a practical matter, they cannot be physically undone due to a multitude of intermingled dependencies.

By logically undoing active transactions at the start of a read-only transaction, Appellant's invention avoids the need for a transactional clean point (Appellant's specification, paragraph [0076]). However, it is worth noting that logically undoing pending transactions is not a trivial undertaking. For example, the logical undoing of a transaction inserting 1000 records may require a considerable number of operations and may utilize significant memory resources. Also, as performing the logical undo to create the cache view requires the modification of some database blocks, the system needs to hold these modified blocks in the shared cache (i.e., memory) until the read-only transaction commits (Appellant's specification, paragraph [0076]). To handle the possibility of the shared cache overflowing, therefore, the above-described shadow cache (temporary database) is utilized in accordance with Appellant's invention for receiving any overflow. This shadow database serves as a "backing store" for overflow blocks from the shared cache, thereby avoiding the computationally expensive task of rebuilding the cache view once it has overflowed. Traversing log files to reconstruct older versions of the database can be costly in terms of the log file reads required. Once a given block is reconstructed it is kept in the in memory cache. However, if the cache overflows, these reconstructed blocks can be written to a separate volatile temporary shadow database in Appellant's system. Here, the "shadow cache" refers to a very specific cache (implemented using a temporary database) used to support read-only transactions. There is no guarantee of durability for the shadow cache because its contents are only needed for the duration of the read-only transaction(s). It does not serve as a shadow or backup

database for the database itself.

These features of creating and maintaining the cache view based on logically undoing transactions which have begun but have yet to commit are also specifically described in Appellant's claims. For instance, Appellant's claim 1 includes the following claim limitations:

in conjunction with the cache view and the shadow cache, preserving a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit; and performing the logical undo operation in order to reconstruct a transactionally consistent prior version of the given database upon starting the read-only transaction, thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses.

(Appellant's claim 1, emphasis added)

By logically undoing active transactions, Appellant's invention provides for creating and maintaining in memory a view of the database that can be used for performing read-only transactions in a manner that does not block other transactions (e.g., write transactions) being performed against the database. These limitations of logically undoing transactions to create a view of the database at a particular point in time are also recited in Appellant's claim 17, as follows:

A database system capable of restoring databases to a consistent version, the system comprising:  
a log manager module which manages a transaction log of the database system;  
a cache manager module for managing a shared cache that stores database blocks for use by multiple databases and creating a cache view of a given database created using the transaction log of the given database, said cache view being created in response to a read-only transaction of the given database, said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time; wherein the cache manager utilizes a shadow cache for storing any database blocks that overflow said cache view during use of the cache view by the read-only transaction; and a transaction manager module for performing read-only transactions of the database system and which performs, a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit in order to reconstruct, a transactionally consistent prior version of the given database upon starting the read-only transaction, thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses.

(Appellant's claim 17, emphasis added)

Appellant's review of Hayashi and the other prior art of record finds no teaching of the creation of a transactionally consistent prior version of the database supporting read-only uses. In particular, Hayashi does not provide any teaching of logically undoing incomplete transactions in the creation of a transactionally consistent prior version of the database in the manner provided in Appellant's specification and claims. The Examiner references the Hayashi at col. 5, lines 30-46 for the teaching of logically undoing incomplete transactions. However, the referenced portion of Hayashi describes optimizations for writing transactions to disk as follows:

The shadow paging method locks a shared buffer page by page, and when a transaction updates a given page, the method allows no other transactions to share the locked page. Before the updating transaction reaches a commit point, the contents of the page in the middle of updating are written back to an unused page allotted in a disk (i.e., a database). The database contains page data and a table showing relationships between page numbers and locations on the disk. When the contents of the updated page are completely written back to the allotted page on the disk, an original page is shifted to the allotted page in the table, and then a commitment is given to the transaction. A rollback is achieved by simply discarding the allotted page. This shadow paging method resembles the present invention in that it allots an area for storing write back data. This method, however, allots the area on the disk, prohibits the sharing of a page that is being updated, and never considers an improvement in write back performance.

(Hayashi, col. 5, lines 30-46)

The above-referenced teachings of Hayashi do not describe logically undoing active transactions in creating a transactionally consistent view of an entire database for read-only uses (e.g., decision support system (DSS) use, reports, data mining, or the like). Appellant's review of the balance of the Hayashi reference indicates that Hayashi is focused on optimization techniques for reading and writing share blocks. Hayashi describes the making of "copies" of buffers updated by write transactions during their process of nonvolatization. However, the Hayashi buffer "copies" do not constitute a transactionally consistent view of the database at a particular point in time.

The Examiner also references paragraph [0102] of Appellant's specification as admitting that logically undoing a transaction is known in the art. Appellant

acknowledges that logical undo is a concept that is known and used in crash recovery in log-based transaction systems. However, Appellant's invention performs logical undo differently and uses the results for different purposes. In crash recovery the first phase of crash recovery for a log-based system is to go through all the redo records that are needed for crash recovery and apply them to the database in a batch fashion. The crash recovery process then involves physically undoing all the incomplete actions in batch fashion. In contrast, the present invention performs logical undo on demand as needed in creating a cache view for read-only purposes. Also, after the read-only transactions start, the present invention uses physical undo in a different way -- it uses this technique to undo write operations that occur after the read-only transaction started (Appellant's specification, paragraph [0103]). Appellant's invention uses logical undo to create a transactionally consistent prior version (or view) of the database. This view is then maintained in cache for the duration of a read-only transaction(s), so as to enable the read-only transaction to be performed without blocking (or being blocked by) write transactions which are concurrently occurring the database.

### 3. Conclusion

All told, Hayashi provides no teachings of logically undoing transactions so as to create a transactionally consistent view of a database at a given point in time which is maintained in cache and used to support performance of read-only transactions. As Hayashi does not teach or suggest all of the claim limitations of Appellant's independent claims 1 and 17 (and other dependent claims thereof), it is respectfully submitted that the claims distinguish over this reference and that the Examiner's rejection under Section 102 should not be sustained.

### **F. Sixth Ground: Claims 14 and 30 rejected under 35 U.S.C. 103(a)**

#### 1. General

Under Section 103(a), a patent may not be obtained if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. To establish a prima facie case of obviousness under this section, the Examiner must establish: (1) that there is

some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings, (2) that there is a reasonable expectation of success, and (3) that the prior art reference (or references when combined) must teach or suggest all the claim limitations. (See e.g., MPEP 2142). The reference(s) cited by the Examiner fail to meet these conditions.

## 2. Claims 14 and 30

The Examiner has rejected claims 14 and 30 under 35 U.S.C. 103(a) as being obvious over Hayashi (above) in view of in view of U.S. Patent No 5,701,480 to Raz (hereinafter "Raz"). As to these claims, the Examiner continues to rely on Hayashi as substantially teaching Appellant's invention, but acknowledges that Hayashi does not teach "using the back link log records to skip portions of the transaction log that are irrelevant for undoing the read-only transaction" and adds Raz for these teachings (Second Office Action, page 14).

Claims 14 and 30 are dependent upon Appellant's independent claims 1 and 17 and therefore are believed to be allowable for at least the reasons cited above pertaining to the deficiencies of Hayashi in respect to Appellant's invention. As described under Appellant's Fifth Ground of appeal (incorporated by reference herein), Hayashi does not include teachings of logically undoing transactions so as to create a transactionally consistent view of a database at a given point in time which is maintained in cache and used to support performance of read-only transactions. Claims 14 and 30 are also believed to be patentable for the following additional reasons.

As to claim 14, Appellant's intervening claim 13 includes limitations providing that upon occurrence of write operations, back link log records are added to the database's transaction log that serve to link together blocks of the transaction log that pertain to the read-only transaction (Appellant's claim 13). Although the Examiner references Hayashi at col. 3, line 66 to col. 4, line 7 for the corresponding teaching, that portion of Hayashi simply describes that when the contents of a buffer are written to disk the buffer holding information about updates made by the last read or write operation are no longer needed. Respectfully Appellant does not believe that these teachings of Hayashi are at all analogous to Appellant's claim 13 (or the similar limitations of Appellant's claim 29)

Additionally, the limitations of claim 14 further provide that if the read-only transaction must be undone, the back link log records (described in claim 13 above) are used to skip portions of the transaction log that are irrelevant for undoing the read-only transaction. The Examiner acknowledges that Hayashi does not provide these teachings but states that Raz discloses an "undo" recovery mechanism that provides very fast recovery because only the effects of failed transactions must be undone (Second Office Action, page 14). The Examiner states that it would have been obvious to one of ordinary skill in the caching art to combine the teachings of the cited references because Hayashi's teachings would have allowed Raz's method and system to provide a back link log record to skip portions of the transaction log that are irrelevant for undoing the read-only transaction so as to enable users to gain the ability to skip irrelevant portions of the transaction log (Second Office Action, pages 14-15). However, the referenced teachings of Raz do not appear to include the teachings of back link log records comparable to those of Appellant's claimed invention. Instead, the referenced portion of Raz provides as follows:

A considerable amount of processing time, however, is spent flushing updated records to non-volatile state memory and updating the non-volatile snapshot memory when each transaction is committed .... For transactions that update the same records for multiple transactions, and transactions that are short and do not update many pages, a considerable fraction of the processing time is wasted by flushing the updated records to state memory at the end of every transaction.

(Raz, col. 62, lines 3-17)

As illustrated above, Raz discusses the problems inherent in writing each and every change made by a transaction to disk. However, Appellant's review of the above portion of Raz (and the balance of the Raz reference) finds no comparable teaching of back linking records in the transaction log relating to particular read-only transactions so that the read-only transaction may be more quickly undone.

### 3. Conclusion

As described above, Raz includes no teachings which cure the deficiencies of the Hayashi reference as to Appellant's invention. In addition, Hayashi and Raz, either alone or in combination, do not appear to include the specific teachings of back linking

records in the transaction log relating to particular read-only transactions so that the read-only transaction may be more quickly undone as provided in Appellant's claims 14 and 30. For the reasons stated, it is respectfully submitted that Appellant's claims distinguish over the prior art and that the Examiner's rejection under Section 103 should not be sustained.

**G. Seventh Ground: Claims 11 and 27 rejected under 35 U.S.C. 103(a)**

**1. Claims 11 and 27**

The Examiner has rejected claims 11 and 27 under 35 U.S.C. 103(a) as being obvious over Hayashi (above) in view of *The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition*, IEEE Press, 2000 (hereinafter "IEEE").

Claims 11 and 27 are dependent upon Appellant's independent claims 1 and 17 and therefore are believed to be allowable for at least the reasons cited above pertaining to the deficiencies of Hayashi in respect to Appellant's invention which are described in detail under Appellant's Fifth Ground of appeal (and incorporated by reference herein). The IEEE reference does not cure any of these deficiencies of Hayashi as it does not teach anything analogous to Appellant's claimed approach of logically undoing transactions so as to create a transactionally consistent view of a database at a given point in time which is maintained in cache and used to support performance of read-only transactions

**2. Conclusion**

As the combined references do not teach or suggest all of the claim limitations of Appellant's claims, it is respectfully submitted that the claims distinguish over these references and that the rejection under Section 103 is improper.



## 8. CONCLUSION

The present invention greatly improves the ease and efficiency of the task of performing read-only transactions concurrently with other transactions (e.g., write transactions) in database systems. It is respectfully submitted that the present invention, as set forth in the pending claims, sets forth a patentable advance over the art.

In view of the above, it is respectfully submitted that the Examiner's rejections under 35 U.S.C. Section 102 and 103 should not be sustained. If needed, Appellant's undersigned attorney can be reached at 925 465 0361. For the fee due for this Appeal Brief, please refer to the attached Fee Transmittal Sheet. This Appeal Brief is submitted electronically in support of Appellant's Appeal.

Respectfully submitted,

Date: August 13, 2007

/G. Mack Riddle/

G. Mack Riddle; Reg. No. 55,572  
Attorney of Record

925 465 0361  
925 465 8143 FAX

## 9. CLAIMS APPENDIX

1. In a database system employing a transaction log, an improved method for restoring databases to a consistent version, the method comprising:

providing a shared cache storing database blocks for use by multiple databases;

for a read-only transaction of a given database, creating a cache view of a given database using the given database's transaction log, said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time;

creating a shadow cache for storing any database blocks that overflow said cache view during use of the cache view by the read-only transaction;

in conjunction with the cache view and the shadow cache, preserving a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit; and

performing the logical undo operation in order to reconstruct a transactionally consistent prior version of the given database upon starting the read-only transaction, thereby returning a result comprising a transactionally consistent version of the given database supporting read-only uses.

2. The method of claim 1, wherein during occurrence of the read-only transaction any database blocks associated with the cache view are not written from the shared cache to the given database.

3. The method of claim 1, wherein the shadow cache is implemented via a temporary database table.

4. (Canceled)

5. The method of claim 1, wherein the shadow cache is used only in the event the cache view overflows the cache view.

6. The method of claim 1, further comprising:  
providing an allocation bitmap for indicating database blocks in use in the shadow cache.

7. The method of claim 6, further comprising:  
upon completion of the read-only transaction, deleting the shadow cache by updating the allocation bitmap for allocated database blocks.

8. The method of claim 1, wherein the shadow cache comprises a temporary database table including a first column for maintaining a block number of a cache view block having undo/redo records applied to it and a second column for maintaining a block number in a temporary database allocated to save off a modified block from the cache view.

9. (Canceled)

10. The method of claim 1, further comprising:  
upon termination of the read-only transaction, marking the cache view as closed.

11. The method of claim 10, further comprising:  
traversing the shared cache looking for database blocks to purge; and  
purging database blocks from any cache view that has been marked as closed.

12. The method of claim 1, further comprising:  
reusing the cache view created for the read-only transaction for other read-only transactions, which start within a specified period of time following the start of the read-only transaction.

13. The method of claim 1, further comprising:  
detecting the read-only transaction; and  
upon occurrence of write operations, adding back link log records to the

database's transaction log that serve to link together blocks of the transaction log that pertain to the read-only transaction.

14. The method of claim 13, further comprising:

if the read-only transaction must be undone, using the back link log records to skip portions of the transaction log that are irrelevant for undoing the read-only transaction, wherein the back link log records are only generated in the transaction log when there are active read only transactions.

15. A computer-readable medium having processor-executable instructions for performing the method of claim 1.

16. A downloadable set of processor-executable instructions for performing the method of claim 1.

17. A database system capable of restoring databases to a consistent version, the system comprising:

a log manager module which manages a transaction log of the database system;

a cache manager module for managing a shared cache that stores database blocks for use by multiple databases and creating a cache view of a given database created using the transaction log of the given database, said cache view being created in response to a read-only transaction of the given database, said cache view comprising particular database blocks of the shared cache that record a view of a particular version of the database at a given point in time; wherein the cache manager utilizes a shadow cache for storing any database blocks that overflow said cache view during use of the cache view by the read-only transaction; and

a transaction manager module for performing read-only transactions of the database system and which performs, a logical undo operation for the read-only transaction of the given database for logically undoing transactions which have begun but have yet to commit in order to reconstruct, a transactionally consistent prior version of the given database upon starting the read-only transaction, thereby returning a result

comprising a transactionally consistent version of the given database supporting read-only uses.

18. The system of claim 17, wherein during occurrence of the read-only transaction any database blocks associated with the cache view are not written from the shared cache to the given database.

19. The system of claim 17, wherein the shadow cache is implemented via a temporary database table.

20. (Canceled)

21. The system of claim 17, wherein the shadow cache is used only in the event the cache view overflows the cache view.

22. The system of claim 17, wherein said cache manager maintains an allocation bitmap indicating database blocks in use in the shadow cache.

23. The system of claim 22, wherein said cache manager deletes the shadow cache by updating the allocation bitmap for allocated database blocks.

24. The system of claim 17, wherein the shadow cache comprises a temporary database table including a first column for maintaining a block number of a cache view block having undo/redo records applied to it and a second column for maintaining a block number in a temporary database allocated to save off a modified block from the cache views.

25. (Canceled)

26. The system of claim 17, wherein said cache manager marks the cache view as closed, upon termination of the read-only transaction.

27. The system of claim 26, wherein said cache manager traverses the shared cache looking for database blocks to purge, and purges database blocks from any cache view that has been marked as closed.

28. The system of claim 17, wherein said cache manager reuses the cache view created for the read-only transaction for other read-only transactions which start within a specified period of time following the start of the read-only transaction.

29. The system of claim 17, wherein said log manager detects the read-only transaction, and adds back link log records to the transaction log that serve to link together blocks of the transaction log that pertain to the read-only transaction.

30. The system of claim 29, wherein said log manager uses the back link log records to skip portions of the transaction log that are irrelevant for undoing the read-only transaction, wherein the back link log records are only generated in the transaction log when there are active read only transactions.

**10. EVIDENCE APPENDIX**

*This Appeal Brief is not accompanied by an evidence submission under §§ 1.130, 1.131, or 1.132.*

**11. RELATED PROCEEDINGS APPENDIX**

*Pursuant to Appellant's statement under Section 2, this Appeal Brief is not accompanied by any copies of decisions.*